Analisis Potensi Daerah Melalui Metode *Document Clustering* Laporan Pelaksanaan Kegiatan Kuliah Kerja Nyata-Pembelajaran Pemberdayaan Masyarakat

Dyah Mustikasari[1], Teguh Bharata Adji[2], Abdul Kadir[3]
[1,2,3]Jurusan Teknik Elektro dan Teknologi Informasi, Fakultas Teknik, Universitas Gadjah Mada
e-mail: dyah.mustikasari@gmail.com, adji@ugm.ac.id, akadir64@gmail.com

***Abstrak*— Kuliah Kerja Nyata Pembelajaran Pemberdayaan Masyarakat (KKN-PPM) adalah salah satu bentuk pengabdian UGM yang dilakukan oleh mahasiswanya kepada masyarakat secara langsung. Selama melaksanakan KKN, mahasiswa menyusun dan melaksanakan program kegiatan yang berguna bagi pemberdayaan masyarakat. Setelah selesai, mahasiswa diwajibkan membuat laporan tentang pelaksanaan kegiatan yang telah dilakukan. Dokumen laporan pelaksanaan kegiatan KKN yang terkumpul sudah sangat banyak tetapi belum dimanfaatkan secara maksimal. Laporan ini sebenarnya dapat menjadi sebuah sumber infromasi. Salah satu informasi yang bisa digali dari dokumen laporan tersebut adalah informasi tentang potensi daerah lokasi KKN. Penambangan informasi dari dokumen dapat dilakukan dengan *text mining*. Penelitian ini bertujuan untuk menambang informasi tentang potensi daerah dari dokumen laporan pelaksanaan kegiatan KKN-PPM menggunakan salah satu metode pada *text mining*, yaitu *document clustering*. *Clustering* dilakukan dengan dua pendekatan yaitu, STC dan LINGO, menggunakan Carrot2 Workbench. Penggunaan dua algoritma ini dimaksudkan untuk memperoleh perbandingan algoritma yang memberikan hasil lebih baik dalam penggambaran potensi daerah lokasi KKN-PPM UGM. Hasil dari penelitian ini menunjukkan bahwa algoritma LINGO lebih baik dalam memberikan gambaran tentang potensi daerah dibandingkan algoritma STC. LINGO memunculkan label klaster yang bertema potensi daerah lebih banyak dibanding STC. Dari evaluasi pada penelitian ini, LINGO menghasilkan nilai *F-Measure* 70%, dua kali lebih tinggi daripada STC yang hanya 33%.**

***Kata Kunci*— KKN-PPM UGM, *document clustering*, STC, LINGO, Carrot2**

I. PENDAHULUAN

Dalam perannya untuk ikut memajukan bangsa, Universitas Gadjah Mada memiliki tiga program utama yaitu pendidikan, penelitian dan pengabdian kepada masyarakat atau yang dikenal sebagai Tri Dharma Perguruan Tinggi. Kuliah Kerja Nyata Pembelajaran Pemberdayaan Masyarakat (KKN-PPM) menjadi salah satu bentuk pengabdian UGM yang dilakukan oleh mahasiswa kepada masyarakat secara langsung.

Tugas mahasiswa selama KKN adalah membuat dan melaksanakan program kegiatan yang bermanfaat bagi pemberdayaan masyarakat setempat. Setelah menyelesaikan seluruh program kegiatan, mahasiswa diwajibkan membuat laporan pelaksanaan kegiatan KKN-PPM. Laporan ini berisi tentang penjelasan pelaksanaan kegiatan yang telah dilakukan oleh masing-masing mahasiswa.

Kegiatan KKN telah dilaksanakan bertahun-tahun sehingga laporan yang terkumpul juga sudah sangat banyak. Laporan ini sebenarnya bisa menjadi sumber infromasi. Salah satu informasi yang bisa digali dari dokumen laporan tersebut adalah informasi tentang potensi daerah lokasi KKN. Potensi adalah sesuatu hal yang dapat dijadikan sebagai bahan atau sumber yang akan dikelola, baik melalui usaha yang dilakukan manusia maupun yang dilakukan melalui tenaga mesin [1]. Ragam program kegiatan yang dilakukan mahasiswa di suatu lokasi KKN-PPM bisa menjadi indikator potensi yang dimiliki daerah tersebut. Sebagai contoh, kegiatan KKN tentang penyuluhan perikanan atau budidaya jahe di suatu lokasi KKN bisa menjadi indikator bahwa daerah tersebut memiliki potensi sumber daya alam jahe atau perikanan.

Salah satu metode untuk menambang (*mining*) informasi dari sebuah data adalah *data mining*, atau dalam hal ini, *text mining* karena datanya berbentuk dokumen teks. Menambang data adalah memperoleh informasi lebih dalam dari informasi yang tampak pada sebuah data.

Penelitian ini bertujuan untuk menambang informasi tentang potensi daerah dari dokumen laporan pelaksanaan kegiatan KKN-PPM menggunakan salah satu metode pada *text mining*, yaitu *document clustering*. *Clustering* dilakukan dengan dua pendekatan yaitu, STC dan LINGO. Keduanya merupakan algoritma berbasis frasa. Dengan menggunakan dua algortima, hasil klaster yang diperoleh bisa dibandingkan dan dianalisa manakah yang menghasilkan klaster lebih baik dalam penggambaran potensi daerah lokasi KKN-PPM UGM.

Gambaran potensi daerah yang dihasilkan dari clustering dokumen laporan KKN ini, diharapkan dapat bermanfaat bagi pengelola KKN-PPM, sebagai pertimbangan kegiatan KKN selanjutnya, serta pihak terkait (pemerintah setempat, kabupaten, dan lain-lain) dalam pengembangan daerah.

II. *TEXT MINING*

Text mining merupakan cabang dari *data mining*, sehingga kadang disebut sebagai *text data mining*. *Text mining* secara garis besar dapat dikatakan sebagai analisis data teks, yaitu mendapatkan informasi bermanfaat dari sekumpulan data teks. Informasi ini bukan informasi yang sudah eksplisit tertuang dalam teks, tetapi informasi baru yang dapat disarikan dari kumpulan teks itu melalui pola. *Text mining* mampu

memberikan solusi permasalahan pada data tidak berstruktur (*unstructured data*), meliputi pengelompokan, pemrosesan, dan analisis. Untuk melakukan tugas tersebut, text mining mengadopsi teknik dari bidang lain, seperti *Data mining, Information Retrieval,* Statistik dan Matematik, *Machine Learning, Linguistic, Natural Languange Processing,* dan *Visualization.*

A. Document Clustering

Text mining bisa dikerjakan dengan dua cara yaitu, klasifikasi (*classification*) dan pengelompokkan (*clustering*). Untuk klasifikasi, dokumen dikelompokkan ke dalam kategori yang sebelumnya sudah ditentukan terlebih dahulu. Sedangkan pada *clustering*, dokumen dikelompokkan menurut kesamaannya antara satu dokumen dengan yang lain. Dengan *clustering*, kategori kelompok tidak harus ditentukan terlebih dahulu. Kategori, atau sering disebut label kelompok (label *cluster*), akan muncul otomatis dalam proses *clustering*. Cara ini dianggap lebih mudah untuk kumpulan data yang belum diketahui atau belum bisa diperkirakan kategorinya.

Untuk melakukan *clustering*, dibutuhkan algoritma. Banyak algoritma yang dikembangkan untuk *document clustering*, beberapa di antaranya yaitu K-Means, Bisecting K-means, SHOC, STC, dan LINGO.

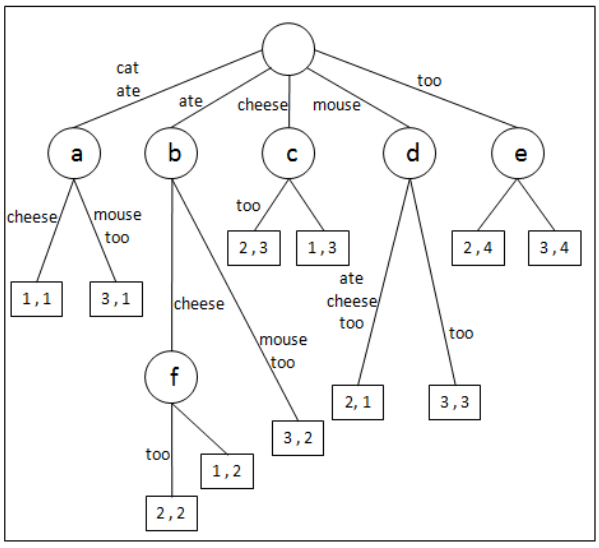
STC (Suffix Tree Clustering)

STC adalah algortima yang menggunakan frasa sehingga prosesnya lebih sederhana dibandingkan dengan algoritma lain. Proses *clustering* dengan STC melalui tiga fase. Fase pertama adalah penguraian dokumen (*document parsing*). Setiap dokumen diubah menjadi urutan kata dan diidentifikasi batas frasanya. Urutan kata bisa diistilahkan sebagai 'kalimat'. Frasa adalah rangkaian kata yang tidak melewati batas frasa [2]. Batas frasa bisa ditandai dengan sintak khusus. Selanjutnya, dokumen dicari kata dasarnya dengan *stemming*.

Tahap selanjutnya adalah identifikasi *cluster* frasa. Pembentukan klaster frasa menggunakan struktur data *suffix tree*. Semua kalimat dalam dokumen dibuat *suffix tree*-nya. Contoh pembentukan *suffix tree* dari kalimat [2]:

1. Cat ate cheese
2. Mouse ate cheese too
3. Cat ate mouse too

Gambar 1 menunjukkan adanya simpul internal yang terbentuk, yaitu a, b, c, d, e, dan f [2]. Simpul internal digambarkan dalam lingkaran, sedangkan kotak berisi angka menggambarkan 'daun'. Simpul internal ini merupakan klaster frasa. Angka pertama pada 'daun' menunjukkan dokumen asal kata/frasa tersebut, sedangkan angka yang kedua menunjukkan posisi kata itu dari *suffix*-nya dimulai. Tabel 1 merangkum hasil klaster frasa dari Gambar 1.



Gambar 1. Pembentukan *Suffix Tree* untuk kalimat "Cat ate cheese","Mouse ate cheese too",dan "Cat ate mouse too"

Tabel 1.
Hasil Klaster Frasa

Simpul	Frasa	Dokumen
A	Cat ate	1,3
B	Ate	1,2,3
C	Cheese	1,2
D	Mouse	2,3
E	Too	2,3
F	Ate cheese	1,2

Setelah *suffix tree* terbentuk, ditentukan simpul mana yang merupakan klaster frasa yang maksimal dan menghitung skor setiap simpul dengan (1). Penghitungan ini merupakan fungsi jumlah dokumen yang terkandung pada tiap simpul dan jumlah frasanya [2].

$$s(m) = |m| \cdot f(|m_p|) \cdot \sum tfidf(w_i) \quad (1)$$

$s(m)$ = skor klaster dasar m
$|m|$ = jumlah dokumen di dalam klaster frasa m
m_p = jumlah kata di dalam m_p yang bukan merupakan *stopword*
w_i = kata-kata di dalam m_p
$tfidf(w_i)$ = nilai yang dihitung untuk setiap m_p

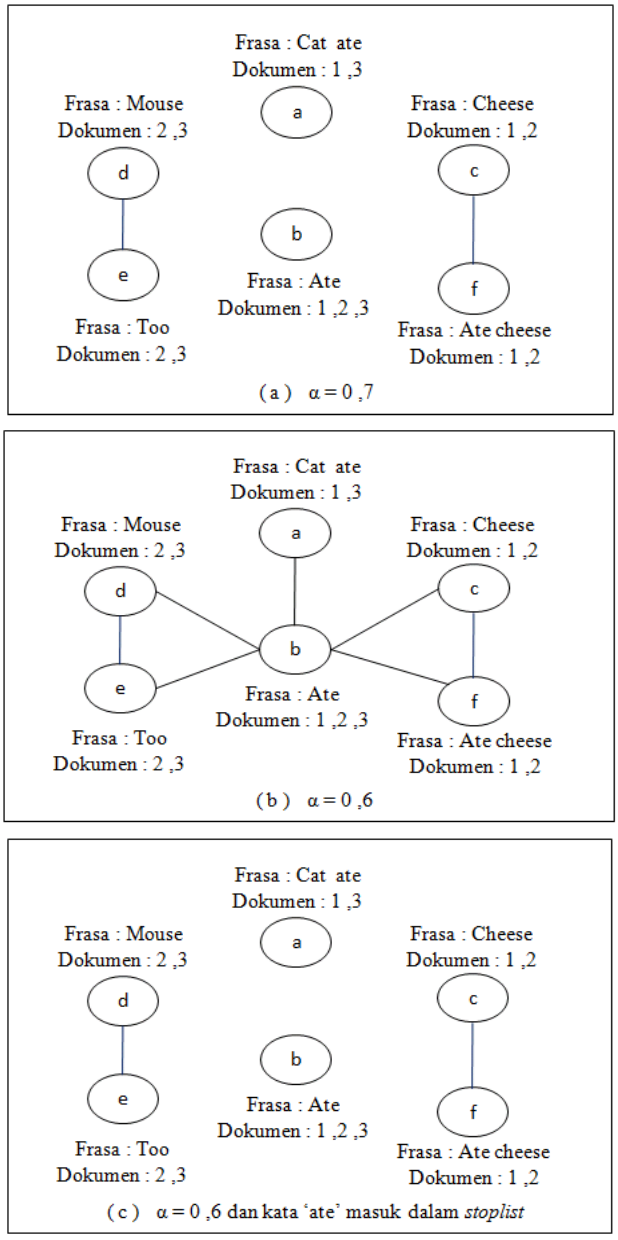
Fungsi f meniadakan frasa dari kata tunggal, linier untuk frasa dengan panjang kata dua sampai enam kata, dan konstan untuk kata yang lebih dari enam kata [2].

Fase terakhir yaitu, penggabungan klaster dasar. Klaster dasar yang terbentuk bisa saja tumpang-tindih atau mirip. Algoritma SCT menggabungkan frasa yang mengandung dokumen yang tumpang-tindihnya tinggi (*high-overlapping*). Sebelum digabungkan, klaster frasa yang tumpang-tindih dihitung kesamaannya. Misalnya terdapat dua klaster frasa m_i dan m_j dengan kesamaan sim(m_i,m_j) adalah:

$$sim(m_i,m_j) = 1 \text{ jika } \quad |m_i \cap m_j| / |m_i| > \alpha, \text{ dan } \; |m_i \cap m_j| / |m_j| > \alpha \quad (2)$$

$sim(m_i,m_j) = 0$ jika yang lainnya

Nilai α disebut *base/phrase cluster merge threshold* atau nilai batas penggabungan klaster frasa. Nilai ini bersifat konstan, dengan nilai antara 0 dan 1. Zamir menggunakan nilai $\alpha = 0.6$ [2]. Klaster-klaster yang terhubung akan digabungkan dan membentuk klaster akhir. Klaster akhir ini beranggotakan semua dokumen dari klaster frasa yang terhubung. Gambar 2 memperlihatkan bahwa nilai α mempengaruhi hasil akhir STC.



Gambar 2. Hasil akhir clustering STC dengan berbagai variasi nilai α

LINGO (Label Induction Group)

LINGO juga merupakan algoritma berbasis frasa. Algoritma ini memiliki beberapa tahap. Tahap pertama adalah *preprocessing* dokumen untuk meningkatkan efisiensi proses clustering. Pada tahap ini dilakukan penghilangan stopword dan proses *stemming*.

Tahap kedua adalah mengekstrak frasa yang sering muncul. Tahap ini memastikan bahwa sebuah frasa yang dipilih adalah lengkap. Frasa lengkap adalah frasa yang tersusun dari kata-kata yang lengkap, sedangkan frasa parsial adalah frasa yang terdiri dari sebagian kata pada frasa lengkap. Cara kerja algoritma ini adalah mengidentifikasi frasa-kiri dan frasa-kanan lalu menggabungkannya menjadi frasa lengkap [3]. Misalnya frasa lengkap "buku cerita anak", maka frasa-kanannya adalah "buku cerita" dan frasa-kirinya adalah "cerita anak".

Setelah frasa komplit terbentuk, tahap ketiga adalah menentukan label klaster. Penentuan ini menggunakan TF-IDF (Term Frequency-Inverse Document Frequency) untuk menghitung bobot setiap frasa [3]. Hasilnya berupa 'matriks term dokumen'. Matrik ini selanjutnya dihitung dengan SVD (*Singular Value Decomposition*) untuk menentukan label klaster.

SVD membagi matriks tersebut menjadi 3 matriks, *V, S, U*. Misalnya matrik itu dinamakan matriks *M*, maka *V, S* dan *U*, adalah $M = USV^T$, yang *S* adalah $t \times d$ "diagonal matriks", *U* adalah $t \times t$ "matriks ortogonal" dan *V* adalah $d \times d$ "matriks ortogonal" [3]. Jumlah nilai singular non-zero matriks *M* merupakan peringkat r_m nya. Untuk kolom matriks *M*, kolom r_m pertama dari matriks *U* membentuk dasar orthogonal. Frasa yang memiliki nilai maksimum dalam vektor dipilih menjadi konsep klaster yang dianggap mudah dipahami oleh pengguna. Selain itu, nilai peringkat kosinus menjadi skor kandidat dari sebuah label cluster.

Taahp keempat pada LINGO adalah menemukan dokumen isi dari setiap klaster yang telah terbentuk. Langkah ini menggunakan VSM (*Vector Space Model*). Misalnya pada sebuah matriks *A*, vektor kolom menunjukkan label klaster. Jika $E = A^T M$, dan *M* adalah matriks 'term dokumen' dari sebuah dokumen. Faktor *eij* dari matriks *E* menunjukkan hubungan dari dokumen ke-*j* terhadap klaster ke-*i*. Jika *eij* melebihi batas nilai (*threshold*), dokumen akan dimasukkan ke dalam klaster. Dokumen yng tidak masuk klaster manapun akan masuk ke klaster "Other Topics" [3].

Langkah terakhir adalah mengetahui klaster final yang memiliki skor maksimum. Skor dihitung dengan mengalikan label skor dengan jumlah anggota klaster. Klaster-klaster akan diurutkan berdasarkan skornya dan ditampilkan kepada pengguna.

III. METODE PENELITIAN

Penelitian ini menggunakan metode *clustering* untuk mengelompokkan dokumen dengan dua algoritma, STC dan LINGO. Keduanya dipilih karena algoritma tersebut berbasis frasa. Algortima ini khusus untuk bidang *text mining*. Berbeda dengan algoritma lain semisal K-Nearest Neighbor (K-NN) atau K-Means yang juga bisa digunakan untuk data terstruktur pada bidang *data mining*.

Penelitian dilakukan dengan beberapa langkah. Langkah pertama adalah persiapan dokumen, kedua adalah *preprocessing*, ketiga *clustering*, dan yang terakhir tahap evaluasi.

A. Data Set

Data yang digunakan dalam penelitian ini adalah dokumen laporan kegiatan KKN-PPM yang dilakukan pada tahun 2012 berlokasi di Kabupaten Sleman. Data tersebut diperoleh dari

bagian penyelenggara kegiatan KKN-PPM yaitu LPPM UGM. Laporan pelaksanaan kegiatan (LPK) KKN-PPM berbentuk narasi deskriptif yang menjelaskan detail setiap program kegiatan KKN yang dilakukan oleh masing-masing mahasiswa. Umumnya, selama melakukan KKN satu orang mahasiswa melaksanakan sekitar sepuluh program kegiatan.

Berkas (*file*) dokumen yang digunakan dalam penelitian ini berjumlah 546 berkas. Berkas ini berbentuk *softcopy* dalam format *Word* (.doc/.docx). Untuk memudahkan proses *clustering*, seluruh dokumen diubah menjadi format *plain text* (.txt) menggunakan *software* ConvertDoc.

B. Preprocessing

Sebelum dilakukan *clustering*, dokumen harus melalui *preprocessing*. Tahap-tahap dalam *preprocessing* yang dilalui pada penelitian ini meliputi:

i. tokenisasi
ii. penghilangan *stopword*
iii. *stemming*

Tokenisasi

Tokenisasi adalah memecah teks menjadi kata tunggal. Kata hasil dari pemecahan ini biasa disebut sebagai token. Token bisa berupa satu kata atau bisa sebuah frasa, misalnya "rumah makan" dapat dipecah per kata menjadi "rumah" dan "makan" atau menjadi satu frasa "rumah makan".

Pada tahap ini juga dilakukan penyeragaman penulisan token, apakah dengan huruf kecil seluruhnya atau huruf kapital. Penyeragaman ini untuk menghindari kesalahan jika proses selanjutnya bersifat *case-sensitive*.

Tanda baca dan angka juga dihilangkan, misalnya tanda titik, tanda tanya, tanda petik, tanda kurung, dan sintak khusus seperti *tag* html. Namun, karena STC dan LINGO merupakan algortima berbasis frasa, tanda titik *tidak* dihilangkan. Tanda baca titik berguna sebagai penanda batas frasa.

Penghilangan Stopword

Stopword merupakan kata yang sering muncul dan dianggap tidak memiliki arti khusus, misalnya "ini", "atau", "dan", "jika", dan sebagainya. Karena dianggap tidak memiliki arti penting pada teks, *stopword* dapat dihilangkan. Pada penelitian ini, *stopword* yang digunakan adalah *stoplist* bahasa Indonesia pada penelitian Tala[4]. Di samping itu, kata-kata yang sering muncul terkait topik bahasan pada dokumen juga masuk dalam *stoplist*. Karena topik bahasan dalam dokumen penelitian ini adalah tentang KKN-PPM, maka kata-kata seperti "laporan", "UGM", "KKN-PPM", "kegiatan", "program", dan sebagainya, juga dihilangkan. *Stoplist* ini diperoleh dari hasil analisis kata-kata yang sering muncul dalam dokumen dan jika dihilangkan tidak menganggu tujuan penelitian yang ingin dicapai.

Karena pada percobaan awal label klaster yang dihasilkan didominasi oleh nama-nama tempat seperti nama dusun, dukuh, desa, dan kecamatan, untuk selanjutnya nama lokasi dihilangkan dari dokumen. Daftar lokasi ini juga dimasukkan dalam *stoplist*.

Penghilangan *stopword* ini akan menyederhanakan teks. Dengan demikian, proses *stemming* lebih cepat karena kata yang harus dicari akar katanya menjadi lebih sedikit.

Stemming

Stemming adalah proses menemukan kata dasar dari sebuah kata turunan. *Stemming* diperlukan dalam *preprocessing* untuk mengurangi tempat penyimpanan istilah dan memperluas arti dari suatu istilah. Bentuk umum kata berimbuhan dalam bahasa Indonesia adalah seperti berikut :

Awalan 1 + Awalan 2 + Kata Dasar + Akhiran 3 + Akhiran 2 + Akhiran 1

Pada bahasa Indonesia, s*temming* memerlukan penghilangan awalan dan akhiran. Beberapa algoritma yang dikembangkan untuk proses *stemming* bahasa Indonesia adalah Algoritma Nazief dan Andriani, algoritma Arifin dan Setiono, algoritma Idris dan Mustofa, Algoritma Ahmad, Yussof, dan Sembok. Penelitian ini menggunakan algoritma Nazief-Andriani.

Algoritma Nazief-Andriani menerapkan aturan berdasarkan pada morfologi bahasa Indonesia. Dibandingkan dengan algortima yang lain, algoritma Nazief-Andriani lebih rumit. Algortima ini juga membutuhkan kamus untuk mengecek kebenaran kata hasil *stemming*. Namun, algoritma ini memiliki performa *stemming* yang paling baik [5]. Dari 3986 kata yang diujikan, algoritma Nazief-Andriani mampu menghasilkan kata dasar dengan benar sebanyak 3714 kata atau 93% dari seluruh data uji [5].

Algoritma Nazief dan Andriani

Aturan imbuhan pada algoritma Nazief dan Adriani adalah sebagai berikut [5]:

[DP + [DP+]] kata dasar [[+DS][+PP][+P]]

DP = *Derivation Prefix* (awalan derivatif)
DS = *Derivation Suffix* (akhiran derivatif)
PP = *Posessive Pronoun* (kata ganti milik)
P = Partikel
Tanda kurung menunjukkan pemberian awalan bersifat opsional.

Secara singkat, tahap-tahap algoritma Nazief dan Adriani adalah sebagai berikut [5]:

1. mencari kata yang akan di-*stem* dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah kata dasar, dan algoritma berhenti.
2. membuang akhiran *inflection* ("-lah", "-kah", "-ku", "-mu", atau "-nya"). Jika berhasil dan akhirannya adalah partikel ("-lah", atau "-kah") maka langkah ini diulangi lagi untuk menghapus imbuhan kepemilikan ("-ku", "-mu", atau "-nya"), jika ada.
3. menghapus akhiran derivatif ("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak, maka dilanjutkan langkah 3a.
   a. Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-k", maka "-k" juga ikut dihapus dan langkah 4 diulangi. Jika tidak berhasil (tidak ditemukan) maka dilakukan langkah 3b.
   b. Akhiran yang dihapus ("-i", "-an" atau "-kan") dikembalikan, dilanjutkan ke langkah 4.
4. menghapus awalan derivatif. Langkah ini memiliki beberapa tahap.
   a. Jika pada langkah 3 ada sufiks yang dihapus, maka langkah selanjutnya adalah memeriksa tabel kombinasi awalan-akhiran yang tidak diijinkan (Tabel 2). Jika ditemukan maka algoritma berhenti.

b. Jika awalan yang ada cocok dengan awalan sebelumnya, maka algoritma berhenti.
c. Jika tiga awalan sudah dihapus maka algoritma berhenti.

Tabel 2.
Kombinasi awalan akhiran yang tidak diperbolehkan

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

Output dari proses *stemming* ini adalah teks yang berisi kata dasar dan kata inti. Teks ini disimpan dalam berkas berbentuk *plain text* dan menjadi hasil akhir dari *preprocessing*.

C. Clustering

Setelah dilakukan *preprocessing*, data siap untuk proses *clustering*. Saat ini telah dikembangkan berbagai macam perangkat untuk lebih memudahkan proses *clustering*. Perangkat bantu *clustering* ini misalnya WEKA, RapidMiner, Clusty, dan lain-lain. Biasanya, setiap perangkat menyediakan beberapa algoritma *clustering*. Pada penelitian ini, perangkat yang digunakan adalah Carrot2 Workbench.

Carrot2

Carrot2 adalah sebuah pustaka dan satu set aplikasi pendukung yang dapat digunakan untuk membangun mesin *clustering*[6]. Carrot2 mampu mengelompokkan hasil pencarian dari mesin pencari semisal Google, Bing, PubMed, dan juga dari sumber dokumen misalnya, Lucene, Apache Solr, dan ElasticSearch [6]. Selain itu, Carrot2 juga mengijinkan dokumen sumber dari direktori internal, dengan syarat berbentuk xml dengan format sebagai berikut.

```
<searchresult>
  <query>query (optional)</query>
    <document>
        <title>Document 1 Title</title>
        <snippet>Document 1 Content.</snippet>
        <url>http://document.url/1</url>
    </document>
    <document>
        <title>Document 2 Title</title>
        <snippet>Document 2 Content.</snippet>
        <url>http://document.url/2</url>
    </document>
    <document>
        <title>Document 3 Title</title>
        <snippet>Document 3 Content.</snippet>
        <url>http://document.url/3</url>
    </document>
</searchresult>
```

Carrot2 dikembangkan dengan bahasa pemrogaman Java, tetapi dapat diintegrasikan dengan kode program dari berbagai bahasa pemrogaman. Untuk diintegrasikan dengan kode program berbasis Java, Carrot2 menyediakan API dan JAR. Untuk bahasa selain Java, pengembang dapat menggunakan Carrot2 *Document Clustering Server* (DCS) dan memanggil Carrot2 *Clustering* menggunakan protokol REST [6]. Carrot2 menyediakan tiga algoritma, Suffix Tree Clustering (STC), LINGO, dan Bisecting K-Means.

Untuk penelitian ini, algoritma yang digunakan hanya STC dan LINGO. Output dari *preprocessing* diubah menjadi bentuk xml sebagaimana format yang telah ditentukan.

D. Evaluasi

Setelah dilakukan *clustering*, cluster yang terbentuk perlu dievaluasi untuk mengetahui seberapa baik *cluster* yang dihasilkan oleh masing-masing algoritma. Banyak pendekatan yang digunakan untuk mengevaluasi kualitas cluster, di antaranya adalah *precision*, *recall*, *F-Measure*, *cluster label quality*, dan *cluster overlap*.

Precision, *Recall*, dan *F-Measure*

Pengukuran kualitas cluster yang dihasilkan oleh beberapa algortima yang berbeda biasanya diukur dengan pendekatan *precision* dan *recall*. Untuk menghitung nilai *precision* dan *recall*, dokumen terlebih dahulu dikategorikan menjadi kelas-kelas 'acuan' atau disebut *ground truth*. Misalnya satu set dokumen *D* dikategorikan menjadi beberapa kelas $C=\{C_1,C_2,C_3,...C_i\}$. *C* menjadi *ground truth*. Algoritma clustering mengelompokkan dokumen *D* menjadi beberapa cluster dengan label $L = \{ L_1, L_2, L_3,...L_i\}$. Nilai *precision* dan *recall* dapat dihitung dengan (3) dan (4). F_i merupakan fungsi dari *precision* dan *recall* [10].

$$prec_i = | L_i \cap C_i | / | L_i | \quad (3)$$

$$rec_i = | L_i \cap C_i | / | C_i | \quad (4)$$

$$F_i = 2 . prec_i . rec_i /(prec_i + rec_i) \quad (5)$$

Namun, sering kali sebuah kelas memiliki kecocokan dengan lebih dari satu klaster atau justru tidak ada cluster yang sesuai dengan kelas manapun, dan sebaliknya (tidak ada kelas yang sesuai dengan cluster manapun). Untuk menghindari *overlap* seperti ini, cluster-cluster yang topiknya serupa dihubungkan hanya pada satu kelas [10].

Cluster Label Quality

Cluster label quality adalah perbandingan antara label yang bermanfaat dengan semua label yang terbentuk [3]. Nilai *cluster label quality* dimulai dari 0.0 (tidak ada cluster yang bermanfaat) hingga 1.0 (semua cluster bermanfaat).

$$q = u/g \quad (6)$$

q = *cluster label quality*
u = banyaknya cluster yang labelnya bermanfaat
g = jumlah semua cluster yang terbentuk

Nilai *q* adalah derajat *usefulness* atas cluster yang terbentuk. Misalnya seorang *user* yang ingin mencari informasi tentang produk elektronik 'Apple' akan menganggap cluster yang berhubungan dengan buah 'Apple' tidak bermanfaat.

Cluster Overlap

Cluster overlap [3] adalah perbandingan jumlah semua dokumen yang ditempatkan di seluruh *cluster* dan semua dokumen (dokumen input).

$$v = a/s - 1 \quad (7)$$

Variabel *a* merupakan jumlah semua dokumen yang ditempatkan di seluruh *cluster*, sedangkan *s* adalah banyaknya dokumen input *clustering*. Pada kasus *overlap*, nilai *a* lebih

besar dari *s*. Nilai *v* paling kecil adalah 0.0, yang artinya tidak ada yang dokumen yang *overlap* dan paling besar adalah tak terhingga.

Nilai *overlap* yang tinggi bisa disebabkan karena sifat datanya. Misalnya data setnya berisi dokumen yang memiliki banyak tema sehingga bisa masuk ke dalam beberapa topik. Meskipun tidak secara langsung menggambarkan kualitas clustering, pengukuran *overlap* bisa membantu dalam analisis hasil evaluasi [10].

IV. HASIL DAN DISKUSI

A. *Hasil Clustering Algoritma STC*

Setelah dilakukan *clustering* menggunakan Carrot2 Workbench, klaster yang dihasikan dari algortima STC sebanyak tiga puluh klaster sebagaimana ditunjukkan pada Tabel 3.

Tabel 3.
Hasil clustering dari algortima STC menggunakan Carrot2

Group id	Klaster	Jumlah Dokumen
0	• Tuju • Lancar • LPK	546
1	• Ramadhan • Masjid • Tpa	419
2	• Sifat Md • Sifat Id	269
3	• Puji Syukur • Rahmat Hidayah	202
4	• Kandang • Kelompok Ternak • Ternak Sapi	227
5	Dosen Bimbing Lapang	151
6	• Lembaga Teliti Abdi Rekapitulasi • Swadaya Pemda	122
7	• Hati • Hari	375
8	Anak Usia	244
9	Tuhan Maha Esa	94
10	Teman Teman	211
11	• Lepas • Hasil Observasi	347
12	Pagi Sore	195
13	Tk Kelola Kembang Umkm	125
14	Rumah Tangga	162
15	Anak Anak Sd	128
16	Sosio Humaniora	191
17	Ajar Anak Anak	108
18	Muda Pemudi	138
19	• Sakit • Obat	242
20	Faktor Faktor Dukung	103
21	Anak Anak Antusias	97
22	• Bimbing Ajar • Ajar Sekolah	139
23	Anak Anak Ikut	95
24	Tingkat Sejahtera	194
25	Anak Anak Ajar	92
26	Suluh Sehat Sehat Gigi	106
27	Tanam Obat	116
28	Sarana	276
29	Baik	274

Carrot2 Workbench menyediakan menu untuk melakukan pengaturan parameter, misalnya nilai α, maksimum frasa/kata tiap label, dan sebagainya. Untuk penelitian ini, nilai α adalah 0,6. Jumlah maksimal klaster frasa adalah 500, klaster frasa setelah urutan tersebut akan mengalami pemotongan. Nilai minimum untuk *s(m)* atau skor klaster dasar adalah 3,00. Klaster yang terbentuk berlabel kata tunggal ataupun frasa, dengan jumlah maksimal frasa adalah tiga dan maksimal kata adalah empat. Jumlah ini sesuai dengan pengaturan awal pada input parameter.

Klaster yang dihasilkan STC ternyata lebih banyak yang berlabel kata umum seperti "Tuju", "Puji Syukur", "Dosen Bimbing Lapang", "Lembaga Teliti Abdi Rekapitulasi" dan seterusnya. Kata-kata ini merupakan kata umum yang terdapat pada laporan KKN-PPM dan sering muncul.

Hasil *clustering* dari algoritma STC ini berkaitan dengan penggunaan *suffix tree* pada proses pembentukan klaster dasar/klaster frasa. Setiap kalimat disusun *suffix tree*-nya mulai dari kata yang paling belakang dan dihitung sampai kata paling depan. Kata yang sering muncul atau frasa yang frekuensinya tinggi akan menempati simpul teratas dan memiliki simpul-simpul anak yang banyak.

Hal ini dapat dijelaskan sebagai berikut. Klaster group id 0 dengan label tiga kata tunggal "Tuju, Lancar, LPK" memiliki anggota terbanyak, yaitu seluruh dokumen, karena kata itu berada di seluruh dokumen dan kemunculannya sering. Misalnya diambil contoh kalimat dari dokumen laporan KKN-PPM yang mengandung kata "Tuju".

1. *tuju* capai tahu dalam pelihara ternak.
2. kk *tuju* informasi sehat gigi mulut .
3. mudah lokal datang cari rumah *tuju* .

Kemudian, ketiga kalimat tersebut dibentuk menjadi *suffix tree*, sebagaimana pada Tabel 4 .

Tabel 4.
Pembentukan *Suffix Tree* untuk kata "Tuju"

Kalimat pertama	Kalimat kedua	Kalimat ketiga
• ternak • pelihara ternak • dalam pelihara ternak • tahu dalam pelihara ternak • capai tahu dalam pelihara ternak • *tuju capai tahu dalam pelihara ternak*	• mulut • gigi mulut • sehat gigi mulut • informasi sehat gigi mulut • *tuju informasi sehat gigi mulut* • kk tuju informasi sehat gigi mulut	• *tuju* • rumah tuju • cari rumah tuju • datang cari rumah tuju • lokal datang cari rumah tuju • mudah lokal datang cari rumah tuju

Dari frasa yang dicetak miring pada setiap kolom, dapat dilihat bahwa kata "Tuju" akan menjadi simpul dari ketiga kalimat tersebut pada *suffix tree*, sedangkan kata lain akan menjadi simpul tunggal saja untuk masing-masing kalimat. Inilah yang menyebabkan kata "Tuju" muncul menjadi label klaster dan beranggotakan seluruh dokumen. Hal ini karena pada setiap dokumen terdapat kata tersebut. Kata lain yang menjadi label misalnya "Ramadhan", "Dosen Bimbing Lapang", "Rumah Tangga", dan seterusnya juga mengalami pembentukan *suffix tree* yang mirip dengan kata "Tuju" sehingga bisa muncul sebagai label klaster. Dari semua klaster yang terbentuk, hanya terdapat satu kalster yang labelnya berkaitan denganpotensi daerah yaitu Group Id 4 dengan label "Kandang, Kelompok Ternak, Ternak Sapi".

B. Hasil Clustering Algoritma LINGO

LINGO menghasilkan klaster yang lebih beragam, sebagaimana yang ditunjukkan pada Tabel 5. Angka dalam tanda kurung pada kolom "Klaster" menunjukkan banyaknya dokumen yang termasuk anggota dalam klaster tersebut. Seperti pada STC, tiap dokumen bisa menjadi anggota dari beberapa klaster.

LINGO menghasilkan label klaster terkait potensi daerah lebih banyak dibandingkan dengan STC. Hal ini karena LINGO membuat terlebih dahulu label klasternya baru kemudian memasukkan dokumen yang sama dengan konsep label tersebut.

Tabel 5.
Hasil *clustering* dari algoritma LINGO menggunakan Carrot2

Group Id	Label Klaster	Group Id	Label Klaster
0	Sifat Md Ternak (99)	31	Sampah Barang (56)
1	Ternak Sapi Kandang (89)	32	Tanam Sayur (56)
2	Gizi Susu (78	33	Wisata Libat (56)
3	Jual Salak (76)	34	Acara Bakti (55)
4	Sifat md hubung (74)	35	Air Sawah (55)
5	Wisata baik (74)	36	Buat Mading (55)
6	Limbah sampah (73)	37	Buku Cerita (55)
7	Siswa siswi kelas sd (73)	38	Damping Paud (55)
8	Pelihara sapi (69)	39	Fakultas Teknologi Tani (55)
9	Pasar Produk Olah ikan (64)	40	Golong Obat (55)
10	Cat warna (63)	41	Latih Guna Software (55)
11	Gigi Mulut Jaga (63)	42	Pasar Internet (55)
12	Faktor Hambat susah (62)	43	Poster Papan (55)
13	Bimbing teknis pasar (58)	44	Sampah Untung (55)
14	Butuh gizi balita (58)	45	Ternak Ajak (55)
15	Foto pasang (58)	46	Usul Bina (55)
16	Ikan Cipta (58)	47	Beli Pakan (54)
17	Plang Kayu (58)	48	Kenal Kambing (54)
18	Pustaka masjid (58)	49	Lahan Toga (54)
19	Bibit Ikan warga (57)	50	Latar Tpa Sarana (54)
20	Buat Kompos Gambar (57)	51	Murid Tk (53)
21	Film Bersih (57)	52	Jahe Jahe (52)
22	Organik Pupuk Kompos (57)	53	Tanam Obat Keluarga Toga Kk (50)
23	Putus Lomba (57)	54	Masalah Hukum (37)
24	Realisasi Awal Sosialisasi (57)	55	Instan Bantu (36)
25	Biogas Sosialisasi (56)	56	Sampah Buah Biogas (33)
26	Butuh Plang Tunjuk Arah (56)	57	Hasil Jambu (24)
27	Gapura Mudah (56)	58	Anggota Uppks (21)
28	Limbah Terna (56)	59	Bioetanol Hasil (14)
29	Mengo Limbah (56)	60	Other Topics (14)
30	Produk Unggul (56)		

LINGO memastikan setiap label merupakan frasa yang komplit. Frasa komplit diperoleh dari kata yang letaknya berdekatan, misalnya kata "Butuh", "Plang", "Tunjuk", dan "Arah". Keempat kata ini muncul berdekatan sehingga dianggap sebagai sebuah frasa komplit. Frekuensi frasa komplit ini dapat diatur pada menu pengaturan. Pada penelitian ini batas frekuensi minimum adalah empat. Frasa yang frekuensinya lebih kecil dari nilai batas (*threshold*) akan diabaikan.

Sayangnya, LINGO menghasilkan klaster "Other Topics". Klaster ini berisi dokumen yang tidak masuk dalam klaster manapun. Namun, karena pada penelitian ini sebuah dokumen bisa menjadi anggota beberapa klaster, maka dokumen yang menjadi anggota klaster "Other Topics" ini sebenarnya juga masuk dalam klaster lain.

C. Perbandingan Hasil Clustering

Setelah diperoleh *cluster* dari algortima STC dan LINGO, cluster yang terbentuk perlu dievaluasi untuk mengetahui kualitas cluster yang dihasilkan oleh masing-masing algoritma. Evaluasi dilakukan dengan mengukur kualitas menggunakan beberapa parameter, yaitu *precision, recall, F-Measure, cluster label quality*, dan *cluster overlap*.

Tabel 6 menunjukkan perbandingan nilai *precision, recall, F-Measure* yang dihasilkan oleh kedua algoritma. Nilai ini diperoleh dengan terlebih dahulu mengelompokkan data set menjadi kelas-kelas bertema potensi daerah. Selanjutnya, klaster dan kelas yang sama topiknya direlasikan dan dihitung *Overall precision, Overall recall*, dan *Overall F-Measure*-nya [10].

Tabel 6.
Perbandingan nilai *precision, recall, F-Measure* dari LINGO dan STC

	Overall Precision	*Overall Recall*	*Overall F-Measure*
LINGO	96%	63%	70%
STC	63%	23%	33%

Nilai *precision, recall, F-Measure* LINGO lebih tinggi daripada STC. *Precision* LINGO cukup tinggi karena hampir seluruh dokumen anggota cluster cocok dengan anggota kelas yang bersesuaian topiknya. *F-Measure* STC jauh lebih rendah dibanding dengan LINGO karena STC hanya menghasilkan sedikit klaster yang bersesuai dengan kelas pada *ground truth*.

Dalam penelitian ini, paramater *cluster label quality* digunakan untuk melihat seberapa besar algoritma dapat memunculkan klaster yang berhubungan dengan potensi daerah. Klaster yang mengandung unsur potensi daerah dinilai memiliki manfaat karena sesuai dengan tujuan *clustering* pada penelitian ini. Sebagaimana yang ditunjukkan Tabel 7, LINGO menghasilkan klaster dengan tema potensi daerah lebih banyak daripada STC sehingga nilai *cluster label quality* lebih besar. Namun, pengukuran ini lebih baik jika *cluster* yang dihasilkan jumlahnya sama.

Tabel 7.
Nilai *cluster label quality* untuk algoritma STC dan LINGO

	u		*g*	***q***
	Group Id	Jumlah		
Lingo	1, 3, 5, 8, 9, 16, 19, 33, 48, 52, 57	11	61	**18%**
STC	4	1	30	**3,3%**

Parameter *overlap* digunakan karena pada penelitian ini sebuah dokumen dimungkinkan masuk ke lebih dari satu

klaster. Tabel 8 menunjukkan nilai *overlap* anggota klaster yang dihasilkan STC dan LINGO.

Tabel 8.
Nilai cluster overlap untuk algoritma STC dan LINGO

	Lingo	STC
a	1601	5894
s	546	546
v	**1,90**	**9,79**

Kedua algoritma menghasilkan nilai overlap yang cukup besar. Hal ini karena setiap data set yang digunakan memiliki tema yang beragam. Satu dokumen laporan kegiatan KKN-PPM dapat memiliki beberapa topik, misalnya pertanian, perikanan, dan wisata karena seorang mahasiswa bisa saja mengerjakan berbagai program kegiatan selama masa KKN-PPM.

V. KESIMPULAN DAN SARAN

Pada penelitian ini, hasil *clustering* algoritma STC kurang bisa memberi gambaran potensi daerah. Hal ini disebabkan kata-kata frekuensi tinggi menjadi simpul *parent* teratas pada pembentukan *suffix tree*, sehingga kata yang mencerminkan potensi daerah tidak bisa muncul menjadi label. Sifat data set yang digunakan pada penelitian ini diperkirakan juga mempengaruhi pembentukan *suffix tree* ini. Topik dari dokumen data set bersifat hampir seragam sehingga membuat algortima membentuk *suffix tree* dengan simpul-simpul kata-kata yang umum di bidang KKN-PPM.

LINGO dapat menghasilkan klaster-klaster yang memberi gambaran potensi daerah. Hal ini karena LINGO menetapkan label klaster terlebih dahulu baru kemudian memasukkan dokumen yang dinilai sesuai dengan konsep klaster ke dalam klaster itu. Algortima LINGO mampu menghasilkan klaster yang baik dan juga dapat dipahami.

Kekurangan algoritma LINGO adalah masih menghasilkan klaster "Other Topics". Oleh karena itu, perlu dilakukan penelitian lebih lanjut untuk mengurangi atau menghilangkan klaster “Others Topics” ini.

DAFTAR PUSTAKA

[1] A.G. Kartasapoetra, Sutedjo, Mulyani, *Teknologi konservasi tanah dan air*, Ed ke-2. Jakarta: Rineka Cipta (1991).

[2] O. E. Zamir, “Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results,” Ph.D thesis, University of Washington, Washington D.C, Amerika Serikat (1999).

[3] S. Osinski, “An Algorithm For Clustering Of Web Search Results.” M.Sc thesis, Poznan University of Technology, Polandia, Juni 2004.

[4] F.Z. Tala, “A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,” Master of Logic Project Institute for Logic, Language and Computation, Universiteit van Amsterdam, Netherlands, (2003).

[5] J. Asian, H. E. Williams, S.M.M. Tahaghoghi, “ Stemming Indonesian,” School of Computer Science and Information Technology, RMIT University, Melbourne, Australia (2005).

[6] S. Osiński, D. Weiss (2004). Carrot2: User and Developer Manual for version 3.9.4 [Online]. Available: http://download.carrot2.org/ stable/ manual/

[7] S. R.Vispute, S. Kanthekar, A. Kadam, C. Kunte, P. Kadam, ”Automatic Personalized Marathi Content Generation,” In *International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, 2014, p.294-299.

[8] S. Osinski, J. Stefanowski, D. Weiss, “Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition,” Poznan University of Technology, Polandia (2004).

[9] S. R.Vispute, M. A. Potey, “Automatic Text Categorization of Marathi Documents Using Clustering Technique,” In *15th International Conference Advanced Computing Technologies (ICACT)*, 2013, p.1-5.

[10] A. Wang, Y. Li, W. Wang, “Text Clustering Based On Key Phrases”, In *The 1st International Conference on Information Science and Engineering*, 2009, p.986 – 989.